IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:	)	Examiner: **Cloud, Joiya M.**
Timothy W. GENSKE, **William G. SWINTON**, **David VOGEL**, **Philippe R. KHAN**, and **Eric O. BODNAR**	)	Art Group: **2144**
Application No. 09/847,811	)	
Filed: May 1, 2001	)	
For: **SYSTEM AND METHOD FOR DYNAMIC UPLOADING AND EXECUTION OF APPLICATIONS AND DRIVERS BETWEEN DEVICES**	)	

Mail Stop Appeal Briefs - Patent
Assistant Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

CORRECTED APPEAL BRIEF

In response to the Notification of Noncompliant Appeal Brief mailed 30 July 2007, Appellants respectfully submit the following Corrected Appeal Brief. No fees are believed to be due. However, Appellants authorize charging of any applicable fees, or refund of any overpayments, to Deposit Account No. 02-2666.

Filed via EFS Web August 30, 2007

TABLE OF CONTENTS

I. REAL PARTY IN INTEREST

LightSurf Technologies Inc. of Santa Cruz, California, a wholly owned subsidiary of Verisign Inc. of Mountain View, California, is the real party in interest.

II. RELATED APPEALS AND INTERFERENCES

To the best of Appellants' knowledge, there are no related appeals or interferences that will directly affect, be directly affected by, or have a bearing on the Board's decision in this appeal.

III. STATUS OF CLAIMS

Claims 1-87 are pending in this application. All claims stand rejected.

IV. STATUS OF AMENDMENTS

Claims 1-3, 5-40 and 42-87 are as originally presented. Claim 41 was previously amended to correct a typographical error, and that amendment was entered. Appellants proposed an amendment to claim 4 that was believed to point out more particularly the material Appellants regard as their invention, but the Examiner declined to enter the amendment. Accordingly, claim 4 is presented in its original form in this appeal.

V. SUMMARY OF CLAIMED SUBJECT MATTER

The invention concerns methodologies for uploading and executing applications and drivers between devices, where a client device probes its environment to identify a host to which it is connected (*see* Abstract; Summary p. 6, ll. 4-6; p. 23, ll. 10-12), then sends (uploads, transmits, injects) an executable driver or application to be executed by the host *(see* Abstract; p. 6, ll. 6-9; p. 23, ll. 12-15). The client invokes execution of the just-uploaded driver or application on the host *(see* Abstract; p. 7, ll. 11-15; p. 39, ll. 17-19), and finally, waits for commands and interactions from the host (*see* Abstract; p. 23, ll. 15-19; p. 43, ll. 2-6).

The specific example of a digital camera being connected to a personal computer ("PC"), uploading a driver to the PC, and responding to commands from the driver is considered extensively as a preferred embodiment (pp. 10-23), including details such as communication protocols that could be used (*e.g.* TCP/IP, p. 24, ll. 10-23) and possible syntaxes for exchanging executable objects, commands and data (*e.g.* Extensible Markup Language or "XML," p. 26, ll. 10-24).

Independent claim 1 recites a method comprising:

connecting a digital camera to a cellular phone capable of hosting the camera (p. 6, ll. 17-19; p. 17, ll. 8-11; p. 41, ll. 26-28; Fig. 4A at 401);

identifying at least one particular cellular phone that is connected to the camera, including determining communication information allowing communication between the camera and the particular cellular phone, and determining command information allowing the camera to invoke execution of a file of interest at the particular cellular phone (p. 6, ll. 4-7 and 19-20; p. 23, ll. 10-15; p. 25, ll. 10-12; p. 41, l. 28 through p. 42, l. 3; Fig. 4A at 402, 403);

based on said determined communication information, transmitting the executable file of interest from said camera to the particular cellular phone (p. 6, ll. 6-9; p. 25, ll. 12-15; p. 38, l. 27 through p. 39, l. 7; Fig. 4A at 406, 407); and

based on said determined command information, invoking execution of the executable file of interest after it has been transmitted to the particular cellular phone (p. p. 7, ll. 11-19; p. 25, l. 15; p. 40, l. 10 through p. 41, l. 22; p. 42, ll. 23-29; Fig. 4B at 409).

Independent claim 41 recites a multi-device system comprising:

a camera (p. 10, l. 23 through p. 18, l. 10; Fig. 1A) that may be connected to a cellular phone that is capable of hosting the camera (p. 10, ll. 6-9); and

a subsystem (p. 25, l. 5 through p. 29, l. 27, Fig. 3), incorporated in the camera, for automatically:

(i) identifying the cellular phone upon connection to the camera, said subsystem initiating communication between the two devices (p. 6, ll. 4-7 and 19-20; p. 23, ll. 10-15; p. 25, ll. 10-12; p. 41, l. 28 through p. 42, l. 3; Fig. 4A at 402, 403);

(ii) uploading the driver of interest from the camera to the cellular phone (p. 6, ll. 6-9; p. 25, ll. 12-15; p. 38, l. 27 through p. 39, l. 7; Fig. 4A at 406, 407); and

(iii) transmitting at least one command from the camera that invokes execution of the driver of interest at the cellular phone, whereupon the driver executes at the cellular phone for controlling operation of the camera (p. p. 7, ll. 11-19; p. 25, l. 15; p. 40, l. 10 through p. 41, l. 22; p. 42, ll. 23-29; Fig. 4B at 409).

Independent claim 51 recites a method for automated transmission, execution, and manipulation of an executable file of interest originating from a first device, upon the first device's connection to a host device, comprising:

connecting the first device to at least one other device capable of hosting the first device (p. 6, ll. 17-19; p. 17, ll. 8-11; p. 41, ll. 26-28; Fig. 4A at 401);

identifying at least one particular host device that is connected to the first device, including determining communication information allowing communication between the first device and the particular host device, and determining command information allowing the first device to manipulate and invoke execution of an executable file of interest at the particular host device (p. 6, ll. 4-7 and 19-20; p. 23, ll. 10-15; p. 25, ll. 10-12; p. 41, l. 28 through p. 42, l. 3; Fig. 4A at 402, 403);

based on said determined communication information, transmitting the executable file of interest from said first device to the particular host device (p. 6, ll. 6-9; p. 25, ll. 12-15; p. 38, l. 27 through p. 39, l. 7; Fig. 4A at 406, 407);

based on said determined command information, transmitting from said first device to the particular host device commands that manipulate the executable file of interest at the particular host device (p. 7, ll. 11-19; p. 25, l. 15; p. 40, l. 10 through p. 41, l. 22; p. 42, ll. 23-29; Fig. 4B at 409); and

initiating a dialog between the two devices (p. 7, l. 19 through p. 8, l. 3), including:

(i) executing said commands transmitted to the host device on the host device (p. 7, ll. 21-25; p. 43, ll. 2-6), and

(ii) in response to said commands transmitted to the host device, returning a reply from the host device to the first device (*e.g.* p. 64, l. 22 through p. 65, l. 29).

Dependent claims 2-5 refine the method of claim 1 to require that the executable file comprise a driver file (p. 7, ll. 11-12;); wherein the driver file controls the operation of the camera (p. 40, ll. 20-23); wherein the executable file comprises a binary file having instructions capable of executing at the cellular phone (p. 7, ll. 3-5; p. 39, ll. 20-22); or wherein the executable file comprises an application capable of executing at the cellular phone (p. 7, ll. 11-12; p. 38, l. 27-p. 39, l. 7).

Dependent claims 11 and 12 discuss embodiments where the camera and cellular phone are permanently or occasionally connected together (p. 6, ll. 18-19).

Dependent claims 13-15 discuss embodiments using various types of serial communication links (p. 20, ll. 14-17).

Dependent claims 68-87 recite specific Extensible Markup Language ("XML") sequences that are described at pages 45-72.

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

The Examiner has rejected claims 1-26, 30, 32, 34, 36-39 and 40 under 35 U.S.C. § 102(e) as anticipated by U.S. Patent No. 6,442,625 issued to Robinson *et al.* ("*Robinson*"). Claims 41-50 and 51-67 are also rejected under § 102(e) "for the same reason." Claims 27-29, 31, 33, 35 and 68-87 stand rejected under 35 U.S.C. § 103(a) as unpatentable over *Robinson* (*supra*) in view of U.S. Patent Application No. 2006/0173781 by Donner *et al.* ("*Donner*").

Appellants seek review of all rejected claims and ask the Board to overturn the Examiner's rejections based on arguments presented in support of independent claim 1, dependent claims 2-5 and 11-15; independent claim 41, independent claim 51; and dependent claims 68-87.

The Examiner has also issued a provisional non-statutory double-patenting rejection of claims 51-67 in view of co-pending U.S. Patent Application No. 09/660,531 by the same inventors. In view of the terminal disclaimer properly filed in that case on 30 August 2007, Appellants respectfully submit that this provisional rejection is moot, and need not be addressed further in this Brief.

VII. ARGUMENT

Appellants will present brief overviews of the main references of record, *Robinson* and *Donner*, then explain why the references are inadequate to support the Examiner's position and why, consequently, the Board should overturn the Examiner's determination and hold that the claims presented are patentable over the prior art of record.

A. Overview of Cited References

1. U.S. Patent No. 6,442,625 to Robinson et al. ("Robinson")

The primary reference relied upon by the Examiner is a U.S. patent granted to Robinson *et al.* for a device containing a flash memory and a standard interface (*e.g.* a digital camera with a PCMCIA interface), and, according to embodiments of the invention, an alternate interface to permit the flash memory data to be transmitted through a cellular phone (*see* Abstract and c. 3, ll. 5-14). A microprocessor in the device (camera) can be provided with an alternate function interface [apparently, driver-like software] downloaded via the standard interface (*see* c. 5, ll. 19-25) so that the camera can output data over the alternate interface to, for example, a GSM cellular phone (*see* c. 5, l. 56 – c. 6, l. 5).

Robinson's system is different from Appellants' claimed invention for several reasons (discussed below), but the claim limitations that are most clearly missing from *Robinson* are the transmission of an executable file from the camera to the cellular phone and invoking the execution of that file. *Robinson* neither transmits nor invokes such an executable file, so the reference fails to support the Examiner's claim rejections under 35 U.S.C. § 102(e).

The Examiner suggests that *Robinson* at c. 6, l. 56 through c. 7, l. 4 and c. 9, ll. 2-4 teach these limitations. However, Appellants respectfully submit that the cited portions of *Robinson* actually describe translating data to be transmitted (*e.g.* a digital photograph) into a GSM format and sending it to the cellular phone in response to a user's manipulation of a user input device. Even though the GSM transmission packet

is said to include "a command," the packet is not an executable file. Furthermore, there is no "invoking" operation.

2. U.S. Patent Application No. 2006/0173781 by Donner ("Donner")

The Examiner relies on *Donner* as a secondary reference in the rejections under 35 U.S.C. § 103(a) of several claims that depend directly or indirectly upon claim 1. *Donner* is a lengthy reference that provides a comprehensive description of a network-based event admittance ticket system. It is largely unrelated to the primary reference, though cellular phones feature prominently and digital cameras are mentioned occasionally in passing, but in any case the reference is relied upon only for relatively undisputed points of network protocol and communication interface operation. *Donner* does not teach or suggest the transmission or invocation of an executable file, and therefore does not cure the defects of the primary reference.

B. Rejection Under 35 U.S.C. § 102(e) Over *Robinson*

1. Claim 1

Independent claim 1 recites a method for automated transmission and execution of an executable file of interest originating from a digital camera, upon the camera's connection to a cellular phone, comprising a number of operations, including transmitting an executable file of interest from said camera to the cellular phone, and invoking execution of the executable file of interest after it has been transmitted to the cellular phone.

(a) No Executable File is Transmitted

The Examiner asserts that the "transmitting" operation is described in *Robinson* at c. 6, l. 56 through c. 7, l. 4 and c. 9, ll. 2-4. Careful review of these sections shows that *Robinson* transmits information from the camera to the phone, but the information is merely that "necessary […] to instruct the cellular phone to dial a preselected telephone number and then transmit the data stored in the flash memory…" Appellants respectfully submit that the information *Robinson* transmits cannot reasonably be described as "an executable file."

It is commonly understood in the art that an executable file is one containing instructions to control the operation of a programmable processor and cause the processor to perform certain functions. The programmable processor is often an electronic device such as a microcontroller that executes machine instructions, but could also be a software interpreter (*e.g.* running on a microcontroller) that performs functions according to interpreter commands.

The information *Robinson* transmits from camera to phone is clearly not a sequence of machine instructions to be executed by the cellular phone – there is nothing in the reference that would suggest such an operational paradigm.

(b) No Transmitted Executable File is Invoked

Claim 1 further recites that, based on previously-determined command information, execution of the executable file of interest is invoked after the file has been transmitted to the cellular phone. In the Examiner's analysis, this operation is allegedly anticipated by the material at c. 7, ll. 1-4, but this material merely concludes the preceding sentence, where the information transmitted to the cellular phone causes the cellular phone to dial a preselected telephone number. In *Robinson*'s system, the bare act of transmitting the "necessary information" to the cellular phone automatically triggers the phone to begin dialing – the camera does not perform an independent "invoking execution" operation, as recited in claim 1.

From a broader perspective, it is clear that *Robinson*'s system operates differently from embodiments of Appellants' invention as described in claim 1. Apart from the most basic similarity (a cellular phone is connected to a digital camera and the devices communicate), each of the allegedly anticipating features of *Robinson* is incongruous with the corresponding claim element. *Robinson* teaches pre-configuring a camera to cause a cellular telephone to dial a number and transmit data when the user pushes a button on the camera, while Appellants' invention is a camera that identifies a cellular phone when a connection is established, including determining communication information allowing communication between the camera and the particular cellular phone, and determining command information allowing the camera to invoke execution of a file of interest at the particular cellular phone; transmitting an executable file of interest to the cellular phone, and invoking execution of the executable file of interest.

For at least the foregoing reasons, the Board should ***overturn*** the Examiner's rejection of claim 1 and hold that this claim is allowable over the prior art of record.

2. Claim 2

The Examiner rejects dependent claims 2-5, which provide additional details about the executable file recited in claim 1, as anticipated by the previously-discussed portions of *Robinson* and an additional portion at c. 9, ll. 1-7. These portions of the reference fail to support the rejections.

Regarding claim 2, the executable file of interest is claimed to be a driver file. As is known in the art, a driver contains executable instructions to control a hardware device. The information *Robinson* transmits does not control the cellular phone; it does not even provide the telephone number the phone is to call. It merely triggers a previously-configured sequence of actions, which are apparently undertaken by hardware and/or software already at the cellular phone, not by instructions in an executable file (driver) transmitted from the camera.

3. Claim 3

Regarding claim 3, the executable file of interest is claimed to control operation of the camera. *Robinson* lacks any sort of control linkage in the phone-to-camera direction – the information transmitted from the camera causes the phone to start performing a preconfigured sequence of operations, but the camera continues to operate autonomously and not under the phone's control.

4. Claim 4

Regarding claim 4, the executable file is claimed to contain instructions capable of executing at the cellular phone. Even if *Robinson*'s transmitted information is (generously) considered to comprise *an instruction* to cause the cellular phone to begin its preconfigured sequence of operations, the information cannot reasonably be described as containing the plurality of instructions required by the plural form of the word "instruction" as used in this claim.

5. Claim 5

Regarding claim 5, the executable file is claimed to comprise an application program capable of executing at the cellular phone. Even acknowledging the broad range of complexity of software that might be described as "an application program," Appellants submit that *Robinson*'s information that merely triggers a preconfigured sequence of operations cannot reasonably be considered to be such a program.

For the foregoing reasons, and further for the reasons discussed in support of claim 1, the Board should ***overturn*** the Examiner's rejection of claims 2-5 and hold that those claims are allowable over the prior art of record.

6. Claims 11 and 12

Claims 11 and 12, which recite wherein the camera and cellular phone are occasionally or permanently connected (respectively), are both rejected over *Robinson*'s Figure 5a, which merely shows camera 503 connected to cell phone 562 by a broken line 560. The reference text associated with Figure 5a is silent on the permanence or impermanence of the connection. Appellants respectfully submit that the Examiner can potentially argue that one type of connection is shown, but not that both types of connection are shown, if the accompanying text does not indicate the connection type.

7. Claims 13-15

The same Figure is also alleged to anticipate claims 13 through 15, which recite particular types of physical phone-to-camera connections. These specific physical interfaces are not mentioned in connection with *Robinson*'s Figure 5a, and two of the claimed interfaces (RS-232 and Universal Serial Bus ("USB")) are not mentioned anywhere in the reference at all.

Appellants have no desire to consume the Board's time and attention addressing every trivial defect or inadvertent oversight in the Examiner's analysis, yet are mindful that arguments not raised in this Brief may be deemed waived. It is hoped that the foregoing material will be sufficient to apprise the Examiner and the Board of the scope of Appellants' objections to the current rejections. Appellants believe that the

deficiencies of *Robinson* with respect to independent claim 1 are serious enough to support a decision in their favor, and would not seek individual consideration of each dependent claim if the independent claim is found to be allowable.

Appellants respectfully request that the Board ***overturn*** the rejections of all the claims specifically discussed above, and of the remaining claims rejected under § 102(e) (6-10, 16-26, 30, 32, 34, 36-39 and 40) at least by virtue of their dependence upon claim 1.

8. Claim 41

Claim 41 recites, in part, a multi-device system wherein a driver is uploaded from a camera to a cellular phone, and at least one command is transmitted from the camera that invokes execution of the driver at the cellular phone. As noted above with respect to the discussion of claim 1, *Robinson* does not teach or suggest transmitting or uploading a driver, or transmitting a command to invoke execution of the driver. Therefore, claim 41, and its dependent claims 42-50 are not anticipated by *Robinson*.

9. Claim 51

Independent claim 51 and some of its dependent claims (52-67) are rejected under 35 U.S.C. § 102(e) as anticipated by *Robinson*. Claim 51 includes the limitations: transmitting an executable file of interest from a first device to a host device; and transmitting host device commands to manipulate the executable file of interest at the host device. These limitations allow claim 51 to benefit from the "no executable file transmitted" and "no executable file invoked" arguments presented above in support of independent claims 1 and 41. Claim 51 also includes the limitation "in response to commands transmitted to the host device, returning a reply from the host device to the first device." The host device and first device seem to correspond to *Robinson*'s cell phone and camera, respectively, but *Robinson* does not teach or suggest returning replies from the cell phone to the camera. The Examiner does not address this limitation of claim 51 directly. Thus, Appellants submit that the Examiner has failed to establish a *prima facie* case of anticipation. The Board should ***overturn*** the rejection of claim 51, and its dependent claims.

C. Rejection Under 35 U.S.C. § 103(a) Over *Robinson* in View of *Donner*

Claims 27-29, 31, 33 and 35 stand rejected under 35 U.S.C. § 103(a) as unpatentable over *Robinson* (*supra*) in view of U.S. Patent Application No. 2006/0173781 by Donner *et al.* ("*Donner*"). These claims depend directly or indirectly upon claim 1 and are patentable over *Robinson* for the reasons discussed above with respect to the independent claim. *Donner* fails to provide the elements missing from *Robinson*.

As noted above, *Robinson* fails to teach or suggest at least the "transmitting an executable file" and "invoking the executable file" claim limitations. The Examiner relies on *Donner* only to establish that various higher-level protocols can operate over sundry lower-level communication links. The mixing and matching of protocols and communication interfaces is not at issue with respect to the independent claims, and careful review of the balance of *Donner* shows that it does not teach "transmitting" or "invoking" either.

Therefore, since *Robinson* and *Donner* in combination do not teach or suggest "transmitting the executable file" or "invoking the file" Appellants respectfully submit that claims 27-29, 31, 33 and 35 are patentable over the combination of Robinson and Donner.

D.* Rejection Under 35 U.S.C. § 103(a) Over *Robinson* in View of *Donner* and *Zintel

1. Claims 68-87

Claims 68-87 stand rejected under 35 U.S.C. § 103(a) as unpatentable over *Robinson* (*supra*) in view of *Donner* (*supra*), and further in view of U.S. Patent No. 6,910,068 to Zintel *et al.* ("*Zintel*"). *Zintel* discusses XML syntax in general, and using XML to exchange identity and capability information between connected devices, but does not teach using XML in most of the situations described in the claim limitations, and does not describe transmitting or invoking executable files. As noted above, neither *Robinson* nor *Donner* teach or suggest the "transmitting an executable file" and "invoking the executable file" limitations recited in independent claim 51, upon which claims 68-87 depend. *Zintel* does not provide support for these limitations either.

Furthermore, with respect to claims 68-87 (which depend upon independent claim 51), none of the references describe the specific XML sequences recited in the claims, or different XML sequences that achieve equivalent results. Therefore, for that additional reason, Appellants respectfully submit that the combination of *Robinson*, *Donner*, and *Zintel* does not make claims 68-87 obvious.

Appellants respectfully submit that the references, alone or in combination do not make the claims as they stand obvious. Therefore, based on the foregoing, Appellants respectfully submit that that the Board should overturn the rejection of claims 1-83 and hold that all of the claims currently under review are allowable.

Respectfully submitted,
BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN, LLP

Dated: August 30, 2007

/James M. Howard/

James M. Howard, Reg. #56,377

VIII. CLAIMS APPENDIX

The claims involved in this appeal are presented below.

1. (Original) In a computer environment where devices are occasionally connected together, a method for automated transmission and execution of an executable file of interest originating from a digital camera, upon the digital camera's connection to a cellular phone, the method comprising:

connecting the digital camera to a cellular phone capable of hosting the camera;

identifying at least one particular cellular phone that is connected to the camera, including determining communication information allowing communication between the camera and the particular cellular phone, and determining command information allowing the camera to invoke execution of a file of interest at the particular cellular phone;

based on said determined communication information, transmitting the executable file of interest from said camera to the particular cellular phone; and

based on said determined command information, invoking execution of the executable file of interest after it has been transmitted to the particular cellular phone.

2. (Original) The method of claim 1, wherein said executable file of interest comprises a driver file.

3. (Original) The method of claim 2, wherein said driver file, upon execution, controls operation of said camera.

4. (Original) The method of claim 1, wherein said executable file comprises a binary file having instructions capable of executing at said cellular phone.

5. (Original) The method of claim 1, wherein said executable file comprises an application program capable of executing at said cellular phone.

6. (Original) The method of claim 1, wherein said camera includes an add-in device capable of being hosted by said cellular phone.

7. (Original) The method of claim 6, wherein said camera comprises a digital camera and wherein said method further comprises:

upon execution of said executable file at said cellular phone, transferring image information from said digital camera to said cellular phone.

8. (Original) At the method of claim 7, further comprising:

after transferring said image information from said digital camera to said cellular phone, wirelessly transmitting said image information to a third device.

9. (Original) The method of claim 1, wherein said cellular phone includes a computing device capable of hosting other devices.

10. (Original) The method of claim 1, wherein said cellular phone includes wireless transmission capability for transferring information received from said camera to other devices.

11. (Original) The method of claim 1, wherein said camera and cellular phones are occasionally connected together.

12. (Original) The method of claim 1, wherein said camera and cellular phones are permanently connected together.

13. (Original) The method of claim 1, wherein said camera and cellular phones are connected together via a serial communication link.

14. (Original) The method of claim 13, wherein said serial communication link comprises an RS-232 serial communication link.

15. (Original) The method of claim 1, wherein said camera and cellular phones are connected together via a USB (Universal Serial Bus) link.

16. (Original) The method of claim 1, wherein invocation of said identifying step occurs upon connecting said camera and cellular phones together.

17. (Original) The method of claim 1, wherein said identifying step includes:

probing the camera's environment for determining which devices, if any, the camera is attached to.

18. (Original) The method of claim 17, wherein said probing step includes:

determining a default communication medium for probing for new devices.

19. (Original) The method of claim 18, wherein said default communication medium is specified initially by factory-preset information.

20. (Original) The method of claim 18, wherein said default communication medium is a selected one of a wireless and a wired communication medium.

21. (Original) The method of claim 20, wherein said default communication medium includes a serial (RS-232) and a USB (Universal Serial Bus) wired communication medium.

22. (Original) The method of claim 19, wherein said factory-preset information is stored in a registry of the camera.

23. (Original) The method of claim 19, wherein said factory-preset information includes a default communication rate and default handshake protocol for at least one potential cellular phone.

24. (Original) The method of claim 17, wherein said probing step includes:

executing an initial sequence of handshake commands and comparing any response received to a list of known responses for identifying a particular cellular phone.

25. (Original) The method of claim 17, wherein said probing step continues until all known potential cellular phones have been enumerated.

26. (Original) The method of claim 1, wherein said identifying step includes:

updating a registry at said camera for indicating any connected cellular phone that has been identified.

27. (Original) The method of claim 1, further comprising:

upon identifying at least one particular cellular phone, ensuring that a state of TCP/IP communication is reached between said camera and the particular identified cellular phone.

28. (Original) The method of claim 27, wherein said step of ensuring that a state of TCP/IP communication is reached includes:

initiating a PPP (Point-to-Point Protocol) communication session between said camera and cellular phones; and, thereafter

initiating a TCP/IP communication session between said camera and cellular phones.

29. (Original) The method of claim 27, wherein said step of ensuring that a state of TCP/IP communication is reached includes:

determining an IP (Internet Protocol) address for said cellular phone.

30. (Original) The method of claim 1, wherein said step of transmitting the executable file of interest includes:

opening the executable file of interest at the camera; and

streaming the opened executable file of interest from the camera to the cellular phone.

31. (Original) The method of claim 30, wherein said streaming step includes:

employing XML protocol for packaging said executable file of interest for delivery to the cellular phone.

32. (Original) The method of claim 30, wherein said step of transmitting further comprises:

returning to said camera a file handle permitting said camera to access said executable file of interest transmitted to said cellular phone.

33. (Original) The method of claim 31, wherein said file handle comprises a file handle that may be understood by said cellular phone for accessing a particular file of interest at said cellular phone.

34. (Original) The method of claim 1, wherein said executable file of interest comprises a byte-code program, and wherein said cellular phone includes capability for executing byte-code programs.

35. (Original) The method of claim 1, wherein said executable file of interest comprises a Java program, and wherein said cellular phone includes a Java Virtual Machine for executing Java programs.

36. (Original) The method of claim 1, wherein said step of invoking execution of the executable file of interest includes:

issuing a command from said camera to said cellular phone to begin execution at said cellular phone of said executable file of interest.

37. (Original) The method of claim 1, wherein said step of invoking execution of the executable file of interest includes:

triggering execution of said executable file indirectly at said cellular phone by instructing said cellular phone to restart itself.

38. (Original) The method of claim 1, further comprising:

placing said camera in a listening mode, after said camera has invoked execution of said executable file at said cellular phone.

39. (Original) The method of claim 38, wherein said camera awaits commands from said cellular phone, while said camera is in a listening mode.

40. (Original) The method of claim 39, wherein commands received at said camera from said cellular phone control operation of said camera.

41. (Previously Presented) A multi-device system providing automated loading and execution of a driver required for connected devices, the system comprising:

a camera that may be connected to a cellular phone that is capable of hosting the camera; and

a subsystem, incorporated in the camera, for automatically:

(i) identifying the cellular phone upon connection to the camera, said subsystem initiating communication between the two devices;

(ii) uploading the driver of interest from the camera to the cellular phone; and

(iii) transmitting at least one command from the camera that invokes execution of the driver of interest at the cellular phone, whereupon the driver executes at the cellular phone for controlling operation of the camera.

42. (Original) The system of claim 41, wherein said driver comprises a binary file having instructions capable of executing at said cellular phone.

43. (Original) The system of claim 42, wherein said binary file comprises native machine instructions for execution by a processor at said cellular phone.

44. (Original) The system of claim 42, wherein said binary file comprises byte-code instructions for execution by an interpreter at said cellular phone.

45. (Original) The system of claim 44, wherein said binary file comprises a Java program, and wherein said cellular phone includes a Java Virtual Machine for executing Java programs.

46. (Original) The system of claim 44, wherein said driver includes:

instructions for unpacking other executable files for execution at said cellular phone.

47. (Original) The system of claim 41, wherein said camera comprises an add-in device capable of being hosted by said cellular phone.

48. (Original) The system of claim 47, wherein said camera comprises a digital camera device, and wherein said cellular phone comprises a handheld device capable of hosting said digital camera device.

49. (Original) The system of claim 48, wherein said handheld computing device functions to retrieve digital image information from said digital camera device and wirelessly transmit that information to another system.

50. (Original) The system of claim 48, wherein said handheld device is a selected one of a cellular phone device and a handheld computing device.

51. (Original) In a computer environment where devices are occasionally connected together, a method for automated transmission, execution, and manipulation of an

executable file of interest originating from a first device, upon the first device's connection to a host device, the method comprising:

connecting the first device to at least one other device capable of hosting the first device;

identifying at least one particular host device that is connected to the first device, including determining communication information allowing communication between the first device and the particular host device, and determining command information allowing the first device to manipulate and invoke execution of an executable file of interest at the particular host device;

based on said determined communication information, transmitting the executable file of interest from said first device to the particular host device;

based on said determined command information, transmitting from said first device to the particular host device commands that manipulate the executable file of interest at the particular host device; and

initiating a dialog between the two devices, including:

(i) executing said commands transmitted to the host device on the host device, and

(ii) in response to said commands transmitted to the host device, returning a reply from the host device to the first device.

52. (Original) The method of claim 51, wherein said commands include a command to load the executable file of interest.

53. (Original) The method of claim 51, wherein said commands include a command to start the executable file of interest.

54. (Original) The method of claim 51, wherein said commands include a command to end the executable file of interest.

55. (Original) The method of claim 51, wherein said commands include a command to activate the executable file of interest.

56. (Original) The method of claim 51, wherein said commands include a command to get the capabilities of the host device.

57. (Original) The method of claim 51, wherein said commands include a command to get a reference to the executable file of interest that is running on the host device.

58. (Original) The method of claim 51, wherein said executable file of interest comprises a Java program, and wherein said host device includes a Java Virtual Machine for executing Java programs.

59. (Original) The method of claim 51, wherein said executable file of interest comprises a byte-code program, and wherein said host device includes capability for executing byte-code programs.

60. (Original) The method of claim 51, further comprising:

placing said host device in a listening mode to receive commands from said first device.

61. (Original) The method of claim 51, further comprising:

after said first device has transmitted a command to said host device, placing said first device in a listening mode to receive a reply transmitted from said host device.

62. (Original) The method of claim 51, wherein said reply transmitted from the host device in response to said command from the first device includes status information.

63. (Original) The method of claim 62, wherein said status information includes error information indicating an execution state of a preceding command executed at the host device.

64. (Original) The method of claim 51, wherein transmission between the devices employs XML protocol.

65. (Original) The method of claim 51, further comprising:

returning to said first device a file handle permitting said first device to access said executable file of interest while it resides at said host device.

66. (Original) The method of claim 65, wherein said file handle comprises a file handle that may be understood by said host device for accessing a particular file of interest at said host device.

67. (Original) The method of claim 51, wherein said dialog includes:

issuing a load application command from said first device to said host device to receive said executable file of interest transmitted from the first device.

68. (Original) The method of claim 67, wherein the load application command is transmitted from the first device to the host device as an XML stream with a syntax of:

```
<LoadApp>
    <name>(app)</name>
    <bin>
        <size>(value)</size>
        (data)
    </bin>
</LoadApp>
```

69. (Original) The method of claim 67, wherein the reply to the load application command is transmitted by the host device to the first device as an XML stream with a syntax of:

```
<LoadAppR>
    <status>(value)</status>
    <handle>(value)</handle>
</LoadAppR>
```

70. (Original) The method of claim 51, wherein said dialog includes:

issuing a release application command from said first device to said host device to be able to delete said executable file of interest.

71. (Original) The method of claim 70, wherein the release application command is transmitted from the first device to the host device as an XML stream with a syntax of:

```
<ReleaseApp>
    <handle>(value)</handle>
</ReleaseApp>
```

72. (Original) The method of claim 70, wherein a reply to the release application command is transmitted by the host device to the first device as an XML stream with a syntax of:

```
<ReleaseAppR>
    <status>(value)</status>
</ReleaseAppR>
```

73. (Original) The method of claim 51, wherein said dialog includes:

issuing a start application command from said first device to said host device to begin execution at said host device of said executable file of interest.

74. (Original) The method of claim 73, wherein the start application command is transmitted from the first device to the host device as an XML stream with a syntax of:

```
<StartApp>
    <handle>(value)</handle>  //Handle to application
</StartApp>
```

75. (Original) The method of claim 73,wherein the reply to the start application command is transmitted by the host device to the first device as an XML stream with the following syntax:

```
<StartAppR>
    <status>(value)</status>  //Standard error replies
</StartAppR>
```

76. (Original) The method of claim 51, wherein said dialog includes:

issuing a stop application command from said first device to said host device to discontinue execution at said host device of said executable file of interest.

77. (Original) The method of claim 76, wherein the stop application command is transmitted from the first device to the host device as an XML stream with a syntax of:

```
<StopApp>
    <handle>(value)</handle>
    <priority>(value)</priority>
</StopApp>
```

78. (Original) The method of claim 76, wherein the reply to the stop application command is transmitted by the host device to the first device as an XML stream with a syntax of:

```
<StopAppR>
    <status>(value)</status>
</StopAppR>
```

79. (Original) The method of claim 51, wherein said dialog includes:

issuing an activate application command from said first device to said host device to bring current execution of said executable file of interest to the forefront at said host device.

80. (Original) The method of claim 79, wherein the activate application command is transmitted from the first device to the host device as an XML stream with a syntax of:

```
<ActivateApp>
    <handle>(value)</handle>
    <priority>(value)</priority>
</ActivateApp>
```

81. (Original) The method of claim 79, wherein a reply to the activate application command is transmitted by the host device to the first device as an XML stream with a syntax of:

```
<ActivateAppR>
    <status>(value)</status>
</ActivateAppR>
```

82. (Original) The method of claim 51, wherein said dialog includes:

issuing a command to get information about device capabilities of said host device.

83. (Original) The method of claim 82, wherein the device capabilities command is transmitted from the first device to the host device as an XML stream with a syntax of:

```
<GetCap>
</GetCap>
```

84. (Original) The method of claim 82, wherein the reply to the get capabilities command is transmitted by the host device to the first device as an XML stream with a syntax of:

```
<GetCapR>
    <status>(value)</status>
    <lang>(value)</lang>
    <id>(value)</id>
    <imei>(value)</imei>
    <imsi>(value)</imsi>
    <screen>(value)</screen>
    <version>(value)</version>
    <dataLink>(value)</dataLink>
    <flash>(value)</flash>
    <cpu>(value)</cpu>
</GetCapR>
```

85. (Original) The method of claim 51, wherein said dialog includes:

issuing a command to get information about an active application handle for the executable file of interest.

86. (Original) The method of claim 85, wherein the command to get information about an active application handle is transmitted from the first device to the host device as an XML stream with a syntax of:

```
<GetActAppHandle>
</GetActAppHandle>
```

87. (Original) The method of claim 85, wherein a reply to the command to get information about an active application handle is transmitted by the host device to the first device as an XML stream with a syntax of:

```
<GetActAppHandleR>
    <status>(value)</status>
    <handle>(value)</handle>
</GetActAppHandleR>
```

IX. EVIDENCE APPENDIX

No other evidence is submitted in connection with this appeal.

X. RELATED PROCEEDINGS APPENDIX

No related proceedings exist.